

# Multilingual context-based pronunciation learning for Text-to-Speech

Giulia Comini, Manuel Sam Ribeiro, Fan Yang, Heereen Shim, Jaime Lorenzo-Trueba

Amazon Alexa, TTS Research

{gcomini, manuerib, fffyang, shimreen, truebaj}@amazon.com

## Abstract

Phonetic information and linguistic knowledge are an essential component of a Text-to-speech (TTS) front-end. Given a language, a lexicon can be collected offline and Grapheme-to-Phoneme (G2P) relationships are usually modeled in order to predict the pronunciation for out-of-vocabulary (OOV) words. Additionally, post-lexical phonology, often defined in the form of rule-based systems, is used to correct pronunciation within or between words. In this work we showcase a multilingual unified front-end system that addresses any pronunciation related task, typically handled by separate modules. We evaluate the proposed model on G2P conversion and other language-specific challenges, such as homograph and polyphones disambiguation, post-lexical rules and implicit diacritization. We find that the multilingual model is competitive across languages and tasks, however, some trade-offs exists when compared to equivalent monolingual solutions.

**Index Terms:** neural front-end, grapheme-to-phoneme, homograph disambiguation, post-lexical rules, multilingual models, text-to-speech.

## 1. Introduction

Traditionally, the front-end of a Text-to-Speech (TTS) system is a cascade of modules that cover specific linguistic tasks. These may include Grapheme-to-Phoneme (G2P) conversion, post-lexical phonology, text normalization, among others. Each of these tasks can either be a collection of hand-crafted rules, (weighted) finite-state transducer solutions, or, eventually, statistical models. This long-established solution addresses tasks independently, introducing dependencies between components that can be detrimental to efficient maintenance.

Recent studies have been focusing on improving one or more front-end components and the main areas of interest are: 1) approaches to address specific tasks (e.g. [1–9] for G2P conversion); 2) the realisation of a unified front-end [10–13]; 3) the use of multilingual knowledge to improve performances [3–6, 12, 14]; 4) zero/low-resource language expansion [6, 9, 15, 16]. Examples of approaches that combine more than one front-end task are [12], which proposes a multilingual neural system for both G2P and text normalization, and [13], which tweaks a neural model trained on the G2P task to learn to effectively perform homograph disambiguation on English data.

Inspired by these efforts, we propose a *multilingual context-based pronunciation model for Text-to-Speech systems*. This is a unified data-driven solution that addresses any pronunciation related task traditionally handled by independent TTS front-end modules.

Given the promising results in multilingual G2P literature [3–6, 12, 16], we use multilingual pronunciation knowledge to

train our system. We leverage pronunciation data not only at word level, like it is traditionally done for G2P conversion [3, 5–9], but also at sentence level, similarly to [4, 10–13]. This allows us to go beyond G2P conversion and set a step towards a fully learnable neural front-end.

We evaluate our model on the G2P task for 24 languages and dialects, at the word and sentence level, comparing against equivalent monolingual baselines. Additionally, we evaluate the system on language-specific pronunciation tasks, such as homograph disambiguation for British and American English, polyphone disambiguation for Mandarin, post-lexical rules for European French, and implicit diacritization for Standard and Saudi Arabian Arabic. We find that the multilingual model achieves competitive results across all tasks. However, we highlight that there exists trade-offs between multilingual and monolingual solutions, likely depending on the task, the language and, most importantly, the data quality and availability.

The paper is structured as follows: Section 2 describes the data, their format, the evaluation methodology and the proposed model architecture; Section 3 presents the results on the G2P task across languages and analyses the above mentioned language-specific challenges. It additionally shows the impact of using either multilingual or monolingual data for all the considered tasks; Section 4 concludes.

## 2. Multilingual pronunciation learning

**Pronunciation data.** We collect pronunciation data for 24 unique locales. The data consist of  $\langle \text{text}, \text{pronunciation} \rangle$  pairs, where *text* can be either a single word or a sentence. The word level data consist of internal pronunciation dictionaries (lexica) across 21 locales. On average, they contain ~314k word pronunciations, ranging from 50k up to 1.2M entries. The lexica are a collection of multiple sources and they include person names, place names, foreign words, and other domain-specific entries. We additionally generate pronunciation at the sentence level for 18 locales, using internally available front-ends. For another 3 locales, Standard Arabic (arb), Saudi Arabian Arabic (ar-sa) and Mandarin (cnm-cn), we use human-annotated sentence level data, but we do not have lexica. Figure 1 shows the amount of word and sentence level data per locale.

We use the X-SAMPA phonetic transcription scheme to represent phonetic information. We include X-SAMPA stress and diacritics markers, but exclude syllable boundaries.

**Test sets creation.** Because our system operates at the word and sentence level, we construct two types of test sets. For the test set at the word level, we apply an 85/5/10 split for train, development and test sets using individual words from the lexica. This test set is created independently for each locale, following two criteria: 1) all the words with the same lemma belong to the

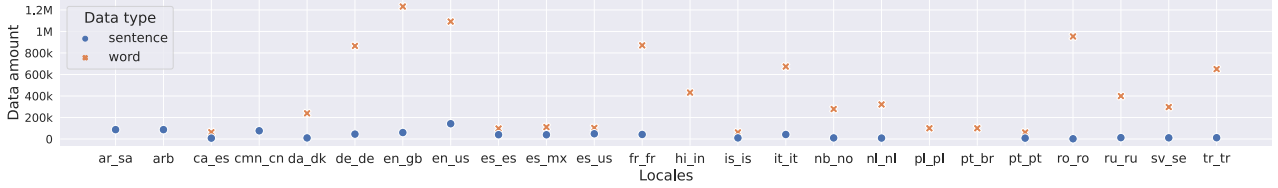


Figure 1: Sentence and word level amount of data. Note that for some locales we only have data for one of the two data types.

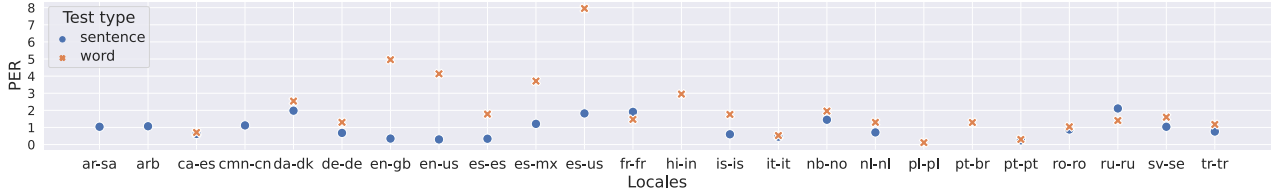


Figure 2: Phone Error Rate (PER%) for the multilingual G2P system at the sentence and word levels.

same train/development/test partition<sup>1</sup>; 2) most of the words in the test set should not be common words in that locale.

Our aim is to have fair evaluation mechanism and to simulate a real-world scenario in which only the most common words are annotated. We rely on the Spacy library [17] to extract word lemmas for the supported languages and on heuristics for the non-supported ones. We get words frequencies information from the multilingual C4 corpus<sup>2</sup> (internally cleaned). We randomly sample the words frequency distribution, limited up to the 95th percentile, and add to the test set all the words with the same lemma of the sampled word. The test set might still contain relatively common words, if they share a lemma with the selected uncommon words. The same procedure is applied for the generation of the development set. On average, the word level test set contains ~30k words per locale.

It is less trivial to design a fair evaluation set at the sentence level. For simplicity, the sentence level test set consists of a random selection of 1-10% of the per-locale sentence data.

**Input/output data format.** In order to handle multilingual information, we prepend languages codes to the input entries, as recommended in [18]. Every locale has a unique code, defined by the W3C language identification tags (ISO 639-1 for the language name and ISO 3166-1 alpha 2 for the country code). The input text is provided at character level, therefore tokenization is not necessary. This allows us to keep a relatively small vocabulary size. For the sentence data, in the absence of punctuation, a word-boundary token is used to represent the space in between words. The target sequences consist of space-separated phonemes in the X-SAMPA format, with the word-boundary token in between words in absence of punctuation. Syllable breaks are discarded. The output vocabulary includes phonemes with X-SAMPA symbols for primary or secondary stress markers, or diacritics, as unique entries.

**Model specifications.** We use a 12-layer transformer encoder-decoder model, as described in [19]. Transformers have been proved successful in sequence-to-sequence tasks, including G2P conversion [4, 5, 7–9, 12]. Leveraging the Open-

NMT [20] implementation of transformer, we train the model for 1M steps on 6 GPUs (Tesla V100-SXM2-32GB), with dynamic batching for up to 4096 tokens. As in [19], we use *Adam* as optimizer and “noam” as learning rate schedule, with *warmup\_steps*=8000. Our model contains ~52.5M parameters.

### 3. Experiments

We train a multilingual G2P model on pronunciation data for 24 locales. We choose the best evaluation step looking at the average performance on the development set across locales. We evaluate the model using the following metrics: word-error-rate (WER%) on the word level test set, sentence-error-rate (SER%) on the sentence level test set and phone-error-rate (PER%) on both test sets. Additionally, we analyse language-specific challenges, such as homograph disambiguation for British and American English (*en-gb*, *en-us*), poly-phone disambiguation for Mandarin (*cmn-cn*), post-lexical rules for French (*fr-fr*) and diacritization for the Arabic locales (*arb* and *ar-sa*). For these locales we also train equivalent monolingual models, in order to understand the impact of the multilingual knowledge on every task.

#### 3.1. Model performances on the G2P task

Figure 2 shows the PERs on the sentence and word level test sets across all locales. On average across locales, the PER on the word level test set is 2.09%, while on the sentence level test set is 0.97%. Higher performances on sentences are expected, since not all words in this test set are out-of-vocabulary (OOV). On average, the WER on the word level test set is 8.45%, while the SER on the sentence level test set is 27.55%. At word level, we obtain the highest PERs on *en-gb*, *en-us*, *es-mx* and *es-us*. The English and Spanish lexica contain foreign and rare domain-specific words, which are challenging and tend to negatively affect the global performance. Overall, we believe that the results on OOV words are competitive, considering that the test set is composed by words with unseen lemmas and that we also generate X-SAMPA stress markers and diacritics.

We compare the proposed multilingual system with equivalent monolingual models for four locales: *cmn-cn*, *en-gb*, *en-us* and *fr-fr*. Table 1 shows that a multilingual model

<sup>1</sup>We opt for this partition following the evaluation design described in the SIGMORPHON 2020 shared task on multilingual G2P conversion [15]

<sup>2</sup><https://www.tensorflow.org/datasets/catalog/c4#c4multilingual>

Table 1: *Monolingual VS multilingual models’ phone, word or sentence error rates (%) on OOV words and sentences on 4 locales.*

	Monolingual		Multilingual	
	Words (PER/WER)	Sentences (PER/SER)	Words (PER/WER)	Sentences (PER/SER)
cmn-cn	/	<b>1.07/31.07</b>	/	1.12/35.47
en-gb	7.86/27.76	0.74/18.87	<b>4.96/18.57</b>	<b>0.35/10.47</b>
en-us	6.74/24.16	0.35/9.02	<b>4.14/16.18</b>	<b>0.30/7.97</b>
fr-fr	3.08/10.01	<b>1.76/40.28</b>	<b>1.47/5.71</b>	1.88/44.04

Table 2: *Homograph disambiguation accuracy (%) for English locales. Results are shown for both the monolingual and multilingual models, at their respective best steps, on a common subset of the homographs evaluation set<sup>3</sup>.*

	Homograph dis. accuracy (%)		
	Monolingual	Multilingual	# words
en-gb	95.49	<b>96.59</b>	821
en-us	94.5	<b>95.88</b>	800

tends to outperform a monolingual one. We observe improvements for en-gb, en-us, and fr-fr on the word level test set. For Mandarin (cmn-cn), the monolingual model outperforms the multilingual one. We hypothesise that Mandarin might not benefit from the other locales knowledge, since it has a different writing system, it doesn’t have word boundaries, it is a tone language and it is highly homographic.

Regarding French, sentence level data poses the additional challenge of post-lexical rules, a phenomenon which we will discuss in Section 3.2.3.

### 3.2. Language-specific analysis

We propose a unified front-end system that is capable of addressing all pronunciation-related tasks, typically handled by separate modules. We analyse the multilingual model performances across a set of languages and tasks, and compare them to equivalent monolingual models. Specifically, we consider homograph disambiguation for English, polyphone disambiguation for Mandarin, post-lexical rules for French, and implicit diacritization for Arabic. We choose these tasks because they are context-dependent and address phenomena that is not trivially handled by word level pronunciation dictionaries.

#### 3.2.1. Homograph disambiguation for English

English homograph disambiguation has more often been approached in literature as a classification problem [10, 21], with recent work exploring it jointly with the G2P task [11, 13]. The latter studies use neural monolingual G2Ps operating at the sentence level: [11] uses a vanilla transformer architecture, but only analyses the accuracy on 3 English homographs, while [13] uses curriculum learning and an homograph disambiguation loss to improve the disambiguation task on top of the G2P task.

We aim to have a fair representation of homographs during training, following [10, 13, 21]. We therefore make use of the open source homograph data released by [10]<sup>3</sup>, consisting of ~14.5k sentences for training and ~1.6k sentences for evaluation. We generate pronunciations at the sentence level for both en-gb and en-us using our internal front-ends. We focus the evaluation on the homograph disambiguation accuracy, comparing the homographs’ gold and predicted X-SAMPA pronunci-

Table 3: *Character pronunciation accuracy (%) for Mandarin. Results are shown for both the monolingual and multilingual models, on the entire sentence and on the polyphonic characters only.*

	Character pronunciation accuracy (%)	
	Whole sentence	Polyphones only
Monolingual	<b>98.80</b>	<b>98.67</b>
Multilingual	98.15	97.24

Table 4: *Phone and word error rates (%) on the French sentence level test set. Results are shown for both the monolingual and multilingual models, on the entire sentence and on the words affected by post-lexical rules.*

	PER/WER	
	Whole sentence	PLRs only
Monolingual	<b>1.76/ -</b>	<b>1.67/6.26</b>
Multilingual	1.88/ -	3.14/8.54

ations. Unfortunately in a number of cases we aren’t able to accurately isolate the phoneme sequences of the homographs in the gold and predicted sequences, therefore we evaluate only on a common subset of the available test data (see Table 2). Table 2 shows that the multilingual model outperforms the respective monolingual model in the homograph disambiguation task for both locales. This is in line with the results of Section 3.1 on the G2P task.

#### 3.2.2. Polyphone disambiguation for Mandarin

Mandarin (cmn-cn) writing system uses a morphosyllabic script, where some characters have a 1:1 relationship with phones, while others, called polyphonic characters, can have multiple pronunciations according to their context. One of the main challenges for Mandarin G2P conversion is polyphone disambiguation (PD). Recent solutions that address the PD task in Mandarin include a unified monolingual neural front-end for joint segmentation, text normalization, part-of-speech prediction, and G2P conversion [22]; a pre-trained Chinese BERT-like [23] language model for polyphone disambiguation [24–27], eventually extended with multi-task training [24, 27].

In this section we show that our system learns PD without employing any additional technique. We extract a list of polyphonic characters from the human-annotated training data. We use ~71.9k sentences for training, and 750 for evaluation. Note that we don’t have a Mandarin lexicon for these experiments. In the test set, out of 15837 characters, 4862 are polyphones (30.7%), with a coverage of 315 unique polyphonic characters.

In Table 3 we compute the character pronunciation accuracy (%) for both the monolingual and multilingual models, and compare it with the PD accuracy (%). Similarly to what we have observed in Section 3.1, the monolingual model outperforms

<sup>3</sup><https://github.com/google/WikipediaHomographData>

Table 5: *Monolingual VS multilingual models’ phone and sentence error rates (%) for Arabic locales, when the input script is diacritized and undiacritized.*

	Monolingual (PER/SER)		Multilingual (PER/SER)	
	with diacritics	w/o diacritics	with diacritics	w/o diacritics
arb	0.46/13.00	<b>1.73/50.00</b>	<b>0.29/9.00</b>	1.85/54.50
ar-sa	0.57/18.50	<b>1.82/51.00</b>	<b>0.32/8.50</b>	<b>1.76/51.50</b>

the multilingual one also in the PD task. Our results cannot be directly compared to others in the recent literature due to differences in the training and test data. However, we do highlight that the PD accuracy is aligned with recent work using more sophisticated neural solutions, often relying on amounts of data from 500k+ [24] to 1M+ [22, 26, 27] training examples.

### 3.2.3. Post-lexical rules for French

French pronunciation relies on complex relationships between words, and phenomena like enchaînement and liaisons are very common. Enchaînement (or linking) consists in taking the last consonant of a word and pronouncing it with the first syllable of the next word. Liaisons determine the pronunciation of latent consonants, e.g. consonants which aren’t pronounced if a word is isolated, but can be pronounced in specific phonetic and syntactic contexts. Taylor et al. [28] confirm the importance of post-lexical rules (PLRs) in French TTS, showing that a French TTS voice is preferred when the front-end uses a PLRs module to correct the pronunciations of words, given a context.

Table 1 compares the performances of monolingual and multilingual models on French OOV words and sentences, but those numbers do not give an indication on how the models perform on PLRs. We therefore evaluate our unified multilingual front-end for post-lexical phenomena on French sentences. We construct a gold set using a set of internal PLRs, derived by a French language expert. Pronunciations are generated at the sentence level, but, for testing purposes, we consider only words which have been modified by PLRs. We evaluate performance in terms of PERs and WERs. Note that both the monolingual and multilingual models use ~33.7k French sentences, while the rest of the French training set consists of ~800k individual words. The test set contains 4211 sentences, 83.16% of which are affected by at least one PLR. Out of 46540 words in the sentence test set, 9877 (21.1%) are affected by the PLRs module.

We previously observed that, for French, a monolingual model outperforms a multilingual one on sentence level inputs (Table 1). The results for PLRs presented in Table 4 show the same trend: the monolingual model obtains 1.67% PER on words affected by PLRs, while the multilingual is at 3.14% PER. These results could be partially explained by the limited amount of entries affected by PLRs in the training data for the multilingual scenario. The multilingual model is likely less capable to learn the nuances which characterise French PLRs. Balancing sentence level and word level data might help alleviate this problem. Additionally, we highlight that we do not have data tailored for the PLRs task and it is possible that the random selection of training sentence might have missed specific contexts. In future work we could design an ad-hoc PLRs set in order to evaluate the model’s ability to generalise, given what it has observed during training.

### 3.2.4. Diacritization for Arabic locales

When Arabic text is diacritized, the G2P task is mostly trivial. However, Arabic text data is often available in the undia-

critized form and a diacritizer becomes a necessary component of the Arabic front-end. Some recent efforts that build Arabic diacritizers rely on the Viterbi algorithm [29, 30] or neural network systems [31–33]. More recently, [34] showed that a transformer model [19] achieves higher performances in the diacritization task when it’s trained in a multi-task fashion to both diacritize and translate. Additionally, [35] proposes to diacritize text using a BERT-like tagger.

In this work, we aim to learn Arabic pronunciation from both diacritized and undiacritized text for two Arabic locales. When the model is given undiacritized input, it must learn the diacritization task. We use the same ~82.2k training sentences for both locales, half of which is undiacritized. We evaluate on a test set of 400 sentences, with a 50% split for diacritized and undiacritized sentences. Note that for both training and testing the content of the diacritized and undiacritized text is the same.

Table 5 shows PER and SER for `arb` and `ar-sa`, comparing the monolingual and the multilingual systems. We observe that both locales benefit from multilingual knowledge when the input text is diacritized. However, when the input text is undiacritized, results are mixed. Overall, in terms of PER, the model performs 3-6 times better when the input is diacritized, which is expected. Although more than half of the undiacritized test sentences contains at least an error, we observe that the undiacritized training sentences are of a relatively limited number (~41.1k) and the PERs are still below 2% for both locales.

## 4. Discussion and conclusions

We present an initial effort towards a fully learnable multilingual neural front-end for TTS. We train a transformer model that generates pronunciations given a word or a normalized text sequence for 24 locales. We demonstrate how the proposed system can effectively predict pronunciation for OOV words with unseen lemmas, as well as learning challenging language-specific tasks such as homograph and polyphone disambiguation, post-lexical phonology and implicit diacritization.

From our analysis it is evident that multilingual knowledge is beneficial for pronunciation learning, however there are trade-offs with respect to equivalent monolingual solutions. We argue that the model performances on a particular task might be affected by the quality and availability of the relevant data for that task. Nonetheless the results are generally promising and set a step towards a unified neural front-end. To the best of our knowledge, this is the first work which gives such a broad overview over a unified TTS front-end.

Future work could involve: 1) incorporating text normalization into the picture; 2) understanding which and how much data are necessary to learn a language pronunciation; 3) exploring low-resource language expansion; 4) using the unified neural front-end to build TTS voices across languages; 5) considering memory and latency requirements for real-time applications.

## 5. References

- [1] M. Bisani and H. Ney, “Joint-sequence models for grapheme-to-phoneme conversion,” *Speech Commun.*, vol. 50, no. 5, pp. 434–451, 2008. [Online]. Available: <https://doi.org/10.1016/j.specom.2008.01.002>
- [2] J. R. Novak, N. Minematsu, and K. Hirose, “WFST-Based Grapheme-to-Phoneme Conversion: Open Source tools for Alignment, Model-Building and Decoding,” in *Proceedings of FSMNLP 2012*. The Association for Computer Linguistics, 2012, pp. 45–49. [Online]. Available: <https://aclanthology.org/W12-6208/>
- [3] B. Peters, J. Dehdari, and J. van Genabith, “Massively Multilingual Neural Grapheme-to-Phoneme Conversion,” in *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, Sep. 2017.
- [4] J. Route, S. Hillis, I. Czeresnia Etinger, H. Zhang, and A. W. Black, “Multimodal, Multilingual Grapheme-to-Phoneme Conversion for Low-Resource Languages,” in *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, Nov. 2019, pp. 192–201.
- [5] M. Yu, H. D. Nguyen, A. Sokolov, J. Lepird, K. M. Sathyendra, S. Choudhary, A. Mouchtaris, and S. Kunzmann, “Multilingual Grapheme-To-Phoneme Conversion with Byte Representation,” in *ICASSP 2020, IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 8234–8238.
- [6] J. Zhu, C. Zhang, and D. Jurgens, “ByT5 model for massively multilingual grapheme-to-phoneme conversion,” in *Interspeech 2022*, 2022, pp. 446–450.
- [7] S. Yolchuyeva, G. Németh, and B. Gyires-Tóth, “Transformer Based Grapheme-to-Phoneme Conversion,” in *Interspeech 2019*.
- [8] S. Wu, R. Cotterell, and M. Huldén, “Applying the Transformer to Character-level Transduction,” in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Apr. 2021, pp. 1901–1907.
- [9] L. Dong, Z.-Q. Guo, C.-H. Tan, Y.-J. Hu, Y. Jiang, and Z.-H. Ling, “Neural Grapheme-To-Phoneme Conversion with Pre-Trained Grapheme Models,” in *ICASSP 2022, IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022, pp. 6202–6206.
- [10] K. Gorman, G. Mazovetskiy, and V. Nikolaev, “Improving homograph disambiguation with supervised machine learning,” in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, May 2018.
- [11] M. Řezáčková, J. Švec, and D. Tihelka, “T5G2P: Using Text-to-Text Transfer Transformer for Grapheme-to-Phoneme Conversion,” in *Proc. Interspeech 2021*, 2021, pp. 6–10.
- [12] A. Conkie and A. Finch, “Scalable Multilingual Frontend for TTS,” in *ICASSP 2020, IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 6684–6688.
- [13] A. Ploujnikov and M. Ravanelli, “SoundChoice: Grapheme-to-Phoneme Models with Semantic Disambiguation,” in *Interspeech 2022*, 2022, pp. 486–490.
- [14] A. Deri and K. Knight, “Grapheme-to-Phoneme Models for (Almost) Any Language,” in *Proceedings of ACL 2016*. The Association for Computer Linguistics, 2016. [Online]. Available: <https://doi.org/10.18653/v1/p16-1038>
- [15] K. Gorman, L. F. Ashby, A. Goyzueta, A. McCarthy, S. Wu, and D. You, “The SIGMORPHON 2020 Shared Task on Multilingual Grapheme-to-Phoneme Conversion,” in *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, Jul. 2020, pp. 40–50.
- [16] X. Li, F. Metze, D. Mortensen, S. Watanabe, and A. Black, “Zero-shot Learning for Grapheme to Phoneme Conversion with Language Ensemble,” in *Findings of the Association for Computational Linguistics: ACL 2022*, May 2022, pp. 2106–2115.
- [17] M. Honnibal and I. Montani, “spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing,” 2017.
- [18] R. Wicks and K. Duh, “The Effects of Language Token Prefixing for Multilingual Machine Translation,” in *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Online only: Association for Computational Linguistics, Nov. 2022, pp. 148–153.
- [19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is All you Need,” in *Advances in Neural Information Processing Systems*, 2017.
- [20] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. Rush, “OpenNMT: Open-Source Toolkit for Neural Machine Translation,” in *Proceedings of ACL 2017, System Demonstrations*, Jul. 2017, pp. 67–72.
- [21] M. Nicolis and V. Klimkov, “Homograph disambiguation with contextual word embeddings for TTS systems,” in *Proc. 11th ISCA Speech Synthesis Workshop (SSW 11)*, Aug. 2021.
- [22] J. Pan, X. Yin, Z. Zhang, S. Liu, Y. Zhang, Z. Ma, and Y. Wang, “A Unified Sequence-to-Sequence Front-End Model for Mandarin Text-to-Speech Synthesis,” in *ICASSP 2020, IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020.
- [23] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*, 2019, pp. 4171–4186.
- [24] Y. Zhang, L. Deng, and Y. Wang, “Unified Mandarin TTS Front-end Based on Distilled BERT Model,” *CoRR*, vol. abs/2012.15404, 2020.
- [25] H. Zhang, “PDF: Polyphone Disambiguation in Chinese by Using FLAT,” in *Proc. Interspeech 2021*, 2021, pp. 4099–4103.
- [26] S. Zhang, K. Zheng, X. Zhu, and B. Li, “A polyphone BERT for Polyphone Disambiguation in Mandarin Chinese,” in *Interspeech 2022*, 2022, pp. 436–440.
- [27] Y. Chen, Y. Steven, Y. Chang, and Y. Yeh, “g2pW: A Conditional Weighted Softmax BERT for Polyphone Disambiguation in Mandarin,” in *Interspeech 2022*, 2022, pp. 1926–1930.
- [28] J. Taylor, S. Maguer, and K. Richmond, “Liaison and pronunciation learning in end-to-end text-to-speech in French,” in *Proc. 11th ISCA Speech Synthesis Workshop (SSW 11)*, Aug. 2021, pp. 195–199.
- [29] K. Darwish, H. Mubarak, and A. Abdelali, “Arabic Diacritization: Stats, Rules, and Hacks,” in *Proceedings of the Third Arabic Natural Language Processing Workshop*, Apr. 2017, pp. 9–17.
- [30] I. Hadjir, M. Abbache, and F. Z. Belkredim, “An Approach for Arabic Diacritization,” in *Natural Language Processing and Information Systems*. Springer International Publishing, 2019.
- [31] A. Fadel, I. Tuffaha, B. Al-Jawarneh, and M. Al-Ayyoub, “Arabic Text Diacritization Using Deep Neural Networks,” *CoRR*, vol. abs/1905.01965, 2019.
- [32] H. Mubarak, A. Abdelali, H. Sajjad, Y. Samih, and K. Darwish, “Highly Effective Arabic Diacritization using Sequence to Sequence Modeling,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jun. 2019, pp. 2390–2395.
- [33] A. Mijlad and Y. E. Younoussi, “Arabic Text Diacritization: Overview and Solution,” in *Proceedings of the 4th International Conference on Smart City Applications*, ser. SCA ’19, 2019.
- [34] B. Thompson and A. Alshehri, “Improving Arabic Diacritization by Learning to Diacritize and Translate,” in *Proceedings of the 19th International Conference on Spoken Language Translation, IWSLT@ACL*, 2022, pp. 11–21.
- [35] W. Nazih, Y. Hifny, and G. Thippa Reddy, “Arabic Syntactic Diacritics Restoration Using BERT Models,” *Intell. Neuroscience*, vol. 2022, Jan 2022.